Workgroup: 6lo
Updates: 8505 (if approved)
Published: 27 April 2020
Intended Status: Standards Track
Expires: 29 October 2020
Authors: P. Thubert, Ed. B. Sarikaya M. Sethi
Cisco Ericsson
R. Struik
Struik Security Consultancy

Address Protected Neighbor Discovery for Low-power and Lossy Networks

Abstract

This document updates the 6LoWPAN Neighbor Discovery (ND) protocol defined in RFC 6775 and RFC 8505. The new extension is called Address Protected Neighbor Discovery (AP-ND) and it protects the owner of an address against address theft and impersonation attacks in a low-power and lossy network (LLN). Nodes supporting this extension compute a cryptographic identifier (Crypto-ID) and use it with one or more of their Registered Addresses. The Crypto-ID identifies the owner of the Registered Address and can be used to provide proof of ownership of the Registered Addresses. Once an address is registered with the Crypto-ID and a proof-of-ownership is provided, only the owner of that address can modify the registration information, thereby enforcing Source Address Validation.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 October 2020.

Table of Contents

1. Introduction

Neighbor Discovery Optimizations for 6LoWPAN networks [RFC6775] (6LoWPAN ND) adapts the original IPv6 Neighbor Discovery (IPv6 ND) protocols defined in [RFC4861] and [RFC4862] for constrained low-power and lossy network (LLN). In particular, 6LoWPAN ND introduces a unicast host Address Registration mechanism that reduces the use

of multicast compared to the Duplicate Address Detection (DAD) mechanism defined in IPv6 ND. 6LoWPAN ND defines a new Address Registration Option (ARO) that is carried in the unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages exchanged between a 6LoWPAN Node (6LN) and a 6LoWPAN Router (6LR). It also defines the Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages between the 6LR and the 6LoWPAN Border Router (6LBR). In LLN networks, the 6LBR is the central repository of all the registered addresses in its domain.

The registration mechanism in "Neighbor Discovery Optimization for Low-power and Lossy Networks" [RFC6775] (aka 6LoWPAN ND) prevents the use of an address if that address is already registered in the subnet (first come first serve). In order to validate address ownership, the registration mechanism enables the 6LR and 6LBR to validate the association between the registered address of a node, and its Registration Ownership Verifier (ROVR). The ROVR is defined in "Registration Extensions for 6LoWPAN Neighbor Discovery" [RFC8505] and it can be derived from the MAC address of the device (using the 64-bit Extended Unique Identifier EUI-64 address format specified by IEEE). However, the EUI-64 can be spoofed, and therefore, any node connected to the subnet and aware of a registered-address-to-ROVR mapping could effectively fake the ROVR. This would allow an attacker to steal the address and redirect traffic for that address. [RFC8505] defines an Extended Address Registration Option (EARO) option that transports alternate forms of ROVRs, and is a pre-requisite for this specification.

In this specification, a 6LN generates a cryptographic ID (Crypto-ID) and places it in the ROVR field during the registration of one (or more) of its addresses with the 6LR(s). Proof of ownership of the Crypto-ID is passed with the first registration exchange to a new 6LR, and enforced at the 6LR. The 6LR validates ownership of the cryptographic ID before it creates any new registration state, or changes existing information.

The protected address registration protocol proposed in this document provides the same conceptual benefit as Source Address Validation (SAVI) [RFC7039] that only the owner of an IPv6 address may source packets with that address. As opposed to [RFC7039], which relies on snooping protocols, the protection is based on a state that is installed and maintained in the network by the owner of the address. With this specification, a 6LN may use a 6LR for forwarding an IPv6 packets if and only if it has registered the address used as source of the packet with that 6LR.

With the 6lo adaptation layer in [RFC4944] and [RFC6282], a 6LN can obtain a better compression for an IPv6 address with an Interface ID (IID) that is derived from a Layer-2 address. As a side note, this

is incompatible with Secure Neighbor Discovery (SeND) [RFC3971] and Cryptographically Generated Addresses (CGAs) [RFC3972], since they derive the IID from cryptographic keys, whereas this specification separates the IID and the key material.

2. Terminology

2.1. BCP 14

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Additional References

The reader may get additional context for this specification from the following references:

* "SEcure Neighbor Discovery (SEND)" [RFC3971],

* "Cryptographically Generated Addresses (CGA)" [RFC3972],

* "Neighbor Discovery for IP version 6" [RFC4861] ,

* "IPv6 Stateless Address Autoconfiguration" [RFC4862], and

* "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals " [RFC4919].

2.3. Abbreviations

This document uses the following abbreviations:

6BBR: 6LoWPAN Backbone Router
6LBR: 6LoWPAN Border Router
6LN: 6LoWPAN Node
6LR: 6LoWPAN Router
CGA: Cryptographically Generated Address
EARO: Extended Address Registration Option
ECDH: Elliptic curve Diffie-Hellman
ECDSA: Elliptic Curve Digital Signature Algorithm
CIPO: Crypto-ID Parameters Option
LLN: Low-Power and Lossy Network
JSON: JavaScript Object Notation
JOSE: JavaScript Object Signing and Encryption
JWK: JSON Web Key
JWS: JSON Web Signature
NA: Neighbor Advertisement

ND: Neighbor Discovery
NDP: Neighbor Discovery Protocol
NDPSO: Neighbor Discovery Protocol Signature Option
NS: Neighbor Solicitation
ROVR: Registration Ownership Verifier
RA: Router Advertisement
RS: Router Solicitation
RSAO: RSA Signature Option
SHA: Secure Hash Algorithm
SLAAC: Stateless Address Autoconfiguration
TID: Transaction ID

3. Updating RFC 8505

Section 5.3 of [RFC8505] introduces the ROVR that is used to detect and reject duplicate registrations in the DAD process. The ROVR is a generic object that is designed for both backward compatibility and the capability to introduce new computation methods in the future. Using a Crypto-ID per this specification is the RECOMMENDED method. Section 7.5 discusses collisions when heterogeneous methods to compute the ROVR field coexist inside a same network.

This specification introduces a new token called a cryptographic identifier (Crypto-ID) that is transported in the ROVR field and used to prove indirectly the ownership of an address that is being registered by means of [RFC8505]. The Crypto-ID is derived from a cryptographic public key and additional parameters.

The overall mechanism requires the support of Elliptic Curve Cryptography (ECC) and of a hash function as detailed in Section 6.2. To enable the verification of the proof, the registering node needs to supply certain parameters including a nonce and a signature that will demonstrate that the node possesses the private-key corresponding to the public-key used to build the Crypto-ID.

The elliptic curves and the hash functions listed in Table 1 in Section 8.2 can be used with this specification; more may be added in the future to the IANA registry. The signature scheme that specifies which combination is used (including the curve and the representation conventions) is signaled by a Crypto-Type in a new IPv6 ND Crypto-ID Parameters Option (CIPO, see Section 4.3) that contains the parameters that are necessary for the proof, a Nonce option ([RFC3971]) and a NDP Signature option (Section 4.4). The NA(EARO) is modified to enable a challenge and transport a Nonce option.

4. New Fields and Options

4.1. New Crypto-ID

The Crypto-ID is transported in the ROVR field of the EARO option and the EDAR message, and is associated with the Registered Address at the 6LR and the 6LBR. The ownership of a Crypto-ID can be demonstrated by cryptographic mechanisms, and by association, the ownership of the Registered Address can be ascertained.

A node in possession of the necessary cryptographic primitives SHOULD use Crypto-ID by default as ROVR in its registrations. Whether a ROVR is a Crypto-ID is indicated by a new "C" flag in the NS(EARO) message.

The Crypto-ID is derived from the public key and a modifier as follows:

1. The hash function used by the signature scheme indicated by the Crypto-Type is applied to the CIPO. Note that all the reserved and padding bits MUST be set to zero.
2. The leftmost bits of the resulting hash, up to the desired size, are used as the Crypto-ID.

At the time of this writing, a minimal size for the Crypto-ID of 128 bits is RECOMMENDED unless backward compatibility is needed [RFC8505]. This value is bound to augment in the future.

4.2. Updated EARO

This specification updates the EARO option to enable the use of the ROVR field to transport the Crypto-ID. The resulting format is as follows:

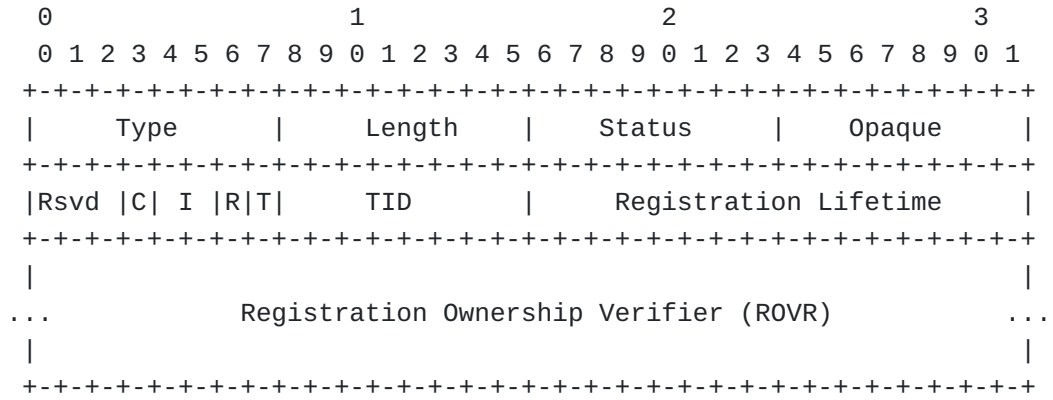

Figure 1: Enhanced Address Registration Option

Type: 33

Length: Defined in [RFC8505] and copied in associated CIPO.

Status: Defined in [RFC8505].

Opaque: Defined in [RFC8505].

Rsvd (Reserved): 3-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

C: This "C" flag is set to indicate that the ROVR field contains a Crypto-ID and that the 6LN MAY be challenged for ownership as specified in this document.

I, R, T: Defined in [RFC8505].

TID: Defined in [RFC8505].

Registration Ownership Verifier (ROVR): When the "C" flag is set, this field contains a Crypto-ID.

This specification uses Status values "Validation Requested" and "Validation Failed", which are defined in [RFC8505].

this specification does not define any new Status value.

4.3. Crypto-ID Parameters Option

This specification defines the Crypto-ID Parameters Option (CIPO). The CIPO carries the parameters used to form a Crypto-ID.

In order to provide cryptographic agility [BCP 201], this specification supports different elliptic-curve based signature schemes, indicated by a Crypto-Type field:

- The ECDSA256 signature scheme, which uses ECDSA with the NIST P-256 curve [FIPS186-4] and the hash function SHA-256 [RFC6234], MUST be supported by all implementations.

- The Ed25519 signature scheme, which uses the Pure Edwards-Curve Digital Signature Algorithm (PureEdDSA) [RFC8032] with the twisted Edwards curve Edwards25519 [RFC7748] and the hash function SHA-512 [RFC6234] internally, MAY be supported as an alternative.

- The ECDSA25519 signature scheme, which uses ECDSA [FIPS186-4] with the Weierstrass curve Wei25519 (see Appendix B.4) and the

hash function SHA-256 [RFC6234], MAY be supported as an alternative.

This specification uses signature schemes that target similar cryptographic strength but rely on different curves, hash functions, signature algorithms, and/or representation conventions. Future specification may extend this to different cryptographic algorithms and key sizes, e.g., to provide better security properties or a simpler implementation.

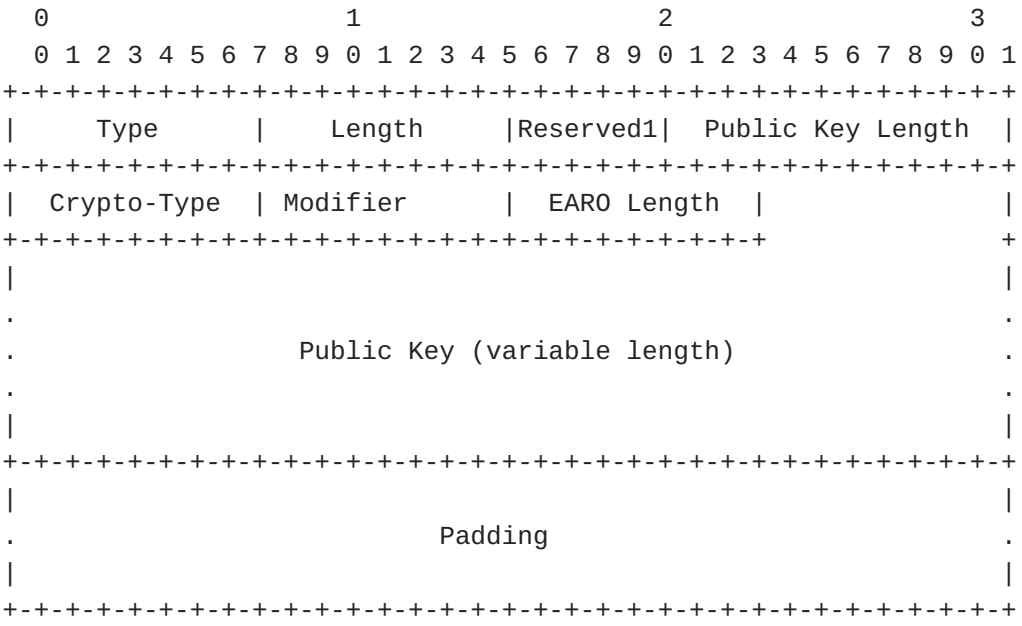

Figure 2: Crypto-ID Parameters Option

Type: 8-bit unsigned integer. to be assigned by IANA, see Table 2.

Length: 8-bit unsigned integer. The length of the option in units of 8 octets.

Reserved1: 5-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

Public Key Length: 11-bit unsigned integer. The length of the Public Key field in bytes. The actual length depends on the Crypto Type and on whether the compressed or uncompressed form is used. The valid values at provided in Table 1.

Crypto-Type: 8-bit unsigned integer. The type of cryptographic algorithm used in calculation Crypto-ID indexed by IANA in the "Crypto-Type Subregistry" in the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" (see Section 8.2).

Modifier:
8-bit unsigned integer. Set to an arbitrary value by the creator of the Crypto-ID. The role of the modifier is to enable the formation of multiple Crypto-IDs from a same key pair, which reduces the traceability and thus improves the privacy of a constrained node that could not maintain many key-pairs.

EARO Length: 8-bit unsigned integer. The option length of the EARO that contains the Crypto-ID associated with the CIPO.

Public Key: A variable-length field, size indicated in the Public Key Length field.

Padding: A variable-length field completing the Public Key field to align to the next 8-bytes boundary. It MUST be set to zero by the sender and MUST be ignored by the receiver.

The implementation of multiple hash functions in a constrained device may consume excessive amounts of program memory. This specification enables the use of the same hash function SHA-256 [RFC6234] for two of the three supported ECC-based signature schemes. Some code factorization is also possible for the ECC computation itself.

[CURVE-REPR] provides information on how to represent Montgomery curves and (twisted) Edwards curves as curves in short-Weierstrass form and illustrates how this can be used to implement elliptic curve computations using existing implementations that already provide, e.g., ECDSA and ECDH using NIST [FIPS186-4] prime curves. For more details on representation conventions, we refer to Appendix B.

4.4. NDP Signature Option

This specification defines the NDP Signature Option (NDPSO). The NDPSO carries the signature that proves the ownership of the Crypto-ID. The format of the NDPSO is illustrated in Figure 3.

As opposed to the RSA Signature Option (RSAO) defined in section 5.2. of SEND [RFC3971], the NDPSO does not have a key hash field. Instead, the leftmost 128 bits of the ROVR field in the EARO are used as hash to retrieve the CIPO that contains the key material used for signature verification, left-padded if needed.

Another difference is that the NDPSO signs a fixed set of fields as opposed to all options that appear prior to it in the ND message that bears the signature. This allows to elide a CIPO that the 6LR

already received, at the expense of the capability to add arbitrary options that would signed with a RSAO.

An ND message that carries an NDPSO MUST have one and only one EARO. The EARO MUST contain a Crypto-ID in the ROVR field, and the Crypto-ID MUST be associated with the keypair used for the Digital Signature in the NDPSO.

The CIPO may be present in the same message as the NDPSO. If it is not present, it can be found in an abstract table that was created by a previous message and indexed by the hash.

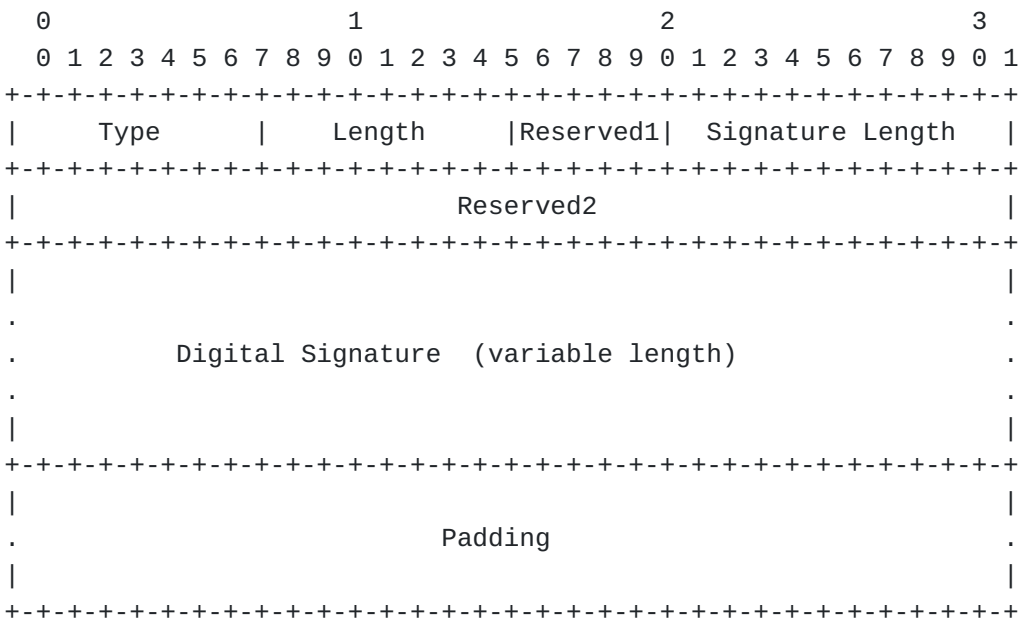

Figure 3: NDP signature Option

Type:
to be assigned by IANA, see Table 2.

Length: 8-bit unsigned integer. The length of the option in units of 8 octets.

Reserved1: 5-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

Digital Signature Length: 11-bit unsigned integer. The length of the Digital Signature field in bytes.

Reserved2: 32-bit unsigned integer. It MUST be set to zero by the sender and MUST be ignored by the receiver.

Digital Signature: A variable-length field containing the digital signature. The length and computation of the digital signature both depend on the Crypto-Type which is found in the associated CIPO, see Appendix B. For the values of the Crypto-Type defined in this specification, and for future values of the Crypto-Type unless specified otherwise, the signature is computed as detailed in Section 6.2.

Padding: A variable-length field completing the Digital Signature field to align to the next 8-bytes boundary. It MUST be set to zero by the sender and MUST be ignored by the receiver.

4.5. Extensions to the Capability Indication Option

This specification defines one new capability bits in the 6CIO, defined by [RFC7400] for use by the 6LR and 6LBR in IPv6 ND RA messages.

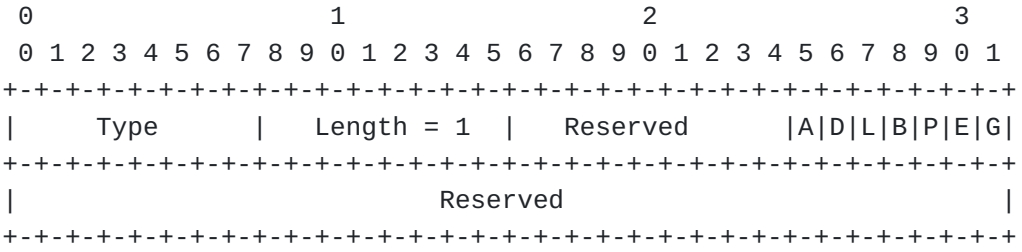

Figure 4: New Capability Bit in the 6CIO

New Option Field:

A:

1-bit flag. Set to indicate that AP-ND is globally activated in the network.

The "A" flag is set by the 6LBR that serves the network and propagated by the 6LRs. It is typically turned on when all 6LRs are migrated to this specification.

5. Protocol Scope

The scope of the protocol specified here is a 6LoWPAN LLN, typically a stub network connected to a larger IP network via a Border Router called a 6LBR per [RFC6775]. A 6LBR has sufficient capability to satisfy the needs of duplicate address detection.

The 6LBR maintains registration state for all devices in its attached LLN. Together with the first-hop router (the 6LR), the 6LBR assures uniqueness and grants ownership of an IPv6 address before it can be used in the LLN. This is in contrast to a traditional network that relies on IPv6 address auto-configuration [RFC4862], where there is no guarantee of ownership from the network, and each IPv6 Neighbor Discovery packet must be individually secured [RFC3971].

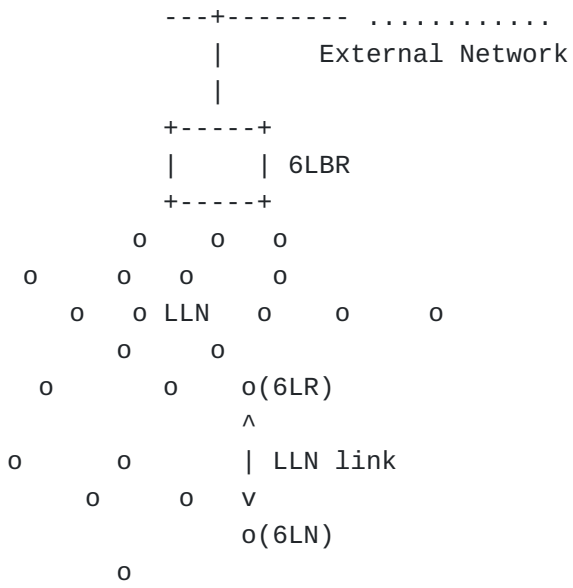

Figure 5: Basic Configuration

In a mesh network, the 6LR is directly connected to the host device. This specification mandates that the peer-wise layer-2 security is deployed so that all the packets from a particular host are securely identifiable by the 6LR. The 6LR may be multiple hops away from the 6LBR. Packets are routed between the 6LR and the 6LBR via other 6LRs.

This specification mandates that all the LLN links between the 6LR and the 6LBR are protected so that a packet that was validated by the first 6LR can be safely routed by other on-path 6LRs to the 6LBR.

6. Protocol Flows

The 6LR/6LBR ensures first-come/first-serve by storing the ROVR associated to the address being registered upon the first registration and rejecting a registration with a different ROVR value. A 6LN can claim any address as long as it is the first to make that claim. After a successful registration, the 6LN becomes the owner of the registered address and the address is bound to the ROVR value in the 6LR/6LBR registry.

This specification protects the ownership of the address at the first hop (the edge). Its use in a network is signaled by the "A" flag in the 6CIO. The flag is set by the 6LBR and propagated unchanged by the 6LRs. The "A" flag enables to migrate a network with the protection off and then turn it on globally.

The 6LN places a cryptographic token, the Crypto-ID, in the ROVR that is associated with the address at the first registration, enabling the 6LR to later challenge it to verify that it is the original Registering Node. The challenge may happen at any time at the discretion of the 6LR and the 6LBR. A valid registration in the 6LR or the 6LBR MUST NOT be altered until the challenge is complete.

When the "A" flag is on, the 6LR MUST challenge the 6LN when it creates a binding with the "C" flag set in the ROVR and when a new registration attempts to change a parameter of that binding that identifies the 6LN, for instance its Source Link-Layer Address. The verification protects against a rogue that would steal an address and attract its traffic, or use it as source address.

The 6LR MUST also challenge the 6LN if the 6LBR directly signals to do so, using an EDAC Message with a "Validation Requested" status. The EDAR is echoed by the 6LR in the NA (EARO) back to the registering node. The 6LR SHOULD also challenge all its attached 6LNs at the time the 6LBR turns the "A" flag on in the 6CIO, to detect an issue immediately.

If the 6LR does not support the Crypto-Type, it MUST reply with an EARO Status 10 "Validation Failed" without a challenge. In that case, the 6LN may try another Crypto-Type until it falls back to Crypto-Type 0 that MUST be supported by all 6LRs.

A node may use more than one IPv6 address at the same time. The separation of the address and the cryptographic material avoids the need for the constrained device to compute multiple keys for

multiple addresses. The 6LN MAY use the same Crypto-ID to prove the ownership of multiple IPv6 addresses. The 6LN MAY also derive multiple Crypto-IDs from a same key.

6.1. First Exchange with a 6LR

A 6LN registers to a 6LR that is one hop away from it with the "C" flag set in the EARO, indicating that the ROVR field contains a Crypto-ID. The Target Address in the NS message indicates the IPv6 address that the 6LN is trying to register [RFC8505]. The on-link (local) protocol interactions are shown in Figure 6. If the 6LR does not have a state with the 6LN that is consistent with the NS(EARO), then it replies with a challenge NA (EARO, status=Validation Requested) that contains a Nonce Option (shown as NonceLR in Figure 6).

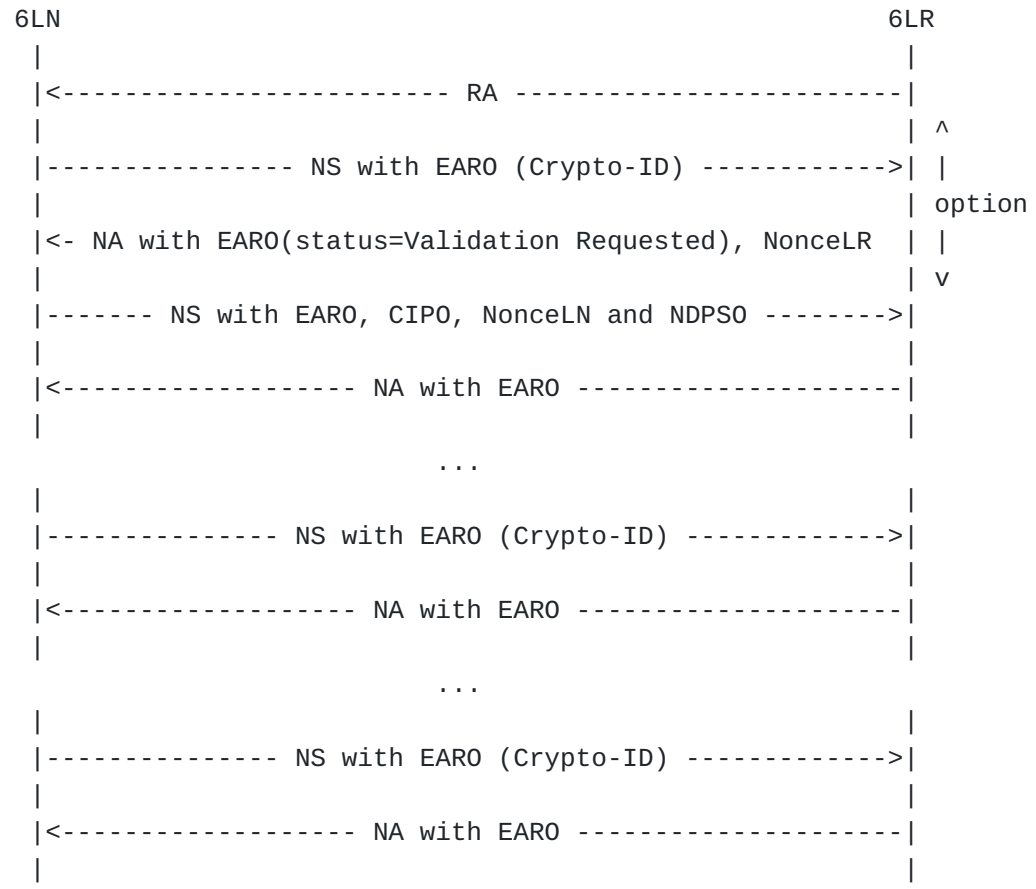

Figure 6: On-link Protocol Operation

The Nonce option contains a nonce value that, to the extent possible for the implementation, was never employed in association with the key pair used to generate the Crypto-ID. This specification inherits from [RFC3971] that simply indicates that the nonce is a random value. Ideally, an implementation uses an unpredictable

cryptographically random value [BCP 106]. But that may be impractical in some LLN scenarios where the devices do not have a guaranteed sense of time and for which computing complex hashes is detrimental to the battery lifetime.

Alternatively, the device may use an always-incrementing value saved in the same stable storage as the key, so they are lost together, and starting at a best effort random value, either as the nonce value or as a component to its computation.

The 6LN replies to the challenge with an NS(EARO) that includes a new Nonce option (shown as NonceLN in Figure 6), the CIPO (Section 4.3), and the NDPSO containing the signature. Both Nonces are included in the signed material. This provides a "contributory behavior", so that either party that knows it generates a good quality nonce knows that the protocol will be secure.

The 6LR MUST store the information associated to a Crypto-ID on the first NS exchange where it appears in a fashion that the CIPO parameters can be retrieved from the Crypto-ID alone.

The steps for the registration to the 6LR are as follows:

Upon the first exchange with a 6LR, a 6LN will be challenged to prove ownership of the Crypto-ID and the Target Address being registered in the Neighbor Solicitation message. When a 6LR receives a NS(EARO) registration with a new Crypto-ID as a ROVR, and unless the registration is rejected for another reason, it MUST challenge by responding with a NA(EARO) with a status of "Validation Requested".

Upon receiving a first NA(EARO) with a status of "Validation Requested" from a 6LR, the registering node SHOULD retry its registration with a Crypto-ID Parameters Option (CIPO) (Section 4.3) that contains all the necessary material for building the Crypto-ID, the NonceLN that it generated, and the NDP signature (Section 4.4) option that proves its ownership of the Crypto-ID and intent of registering the Target Address. In subsequent revalidation with the same 6LR, the 6LN MAY try to omit the CIPO to save bandwidth, with the expectation that the 6LR saved it. If the validation fails and it gets challenged again, then it SHOULD add the CIPO again.

In order to validate the ownership, the 6LR performs the same steps as the 6LN and rebuilds the Crypto-ID based on the parameters in the CIPO. If the rebuilt Crypto-ID matches the ROVR, the 6LN also verifies the signature contained in the NDPSO option. If at that point the signature in the NDPSO option can be verified, then the validation succeeds. Otherwise the validation fails.

If the 6LR fails to validate the signed NS(EARO), it responds with a status of "Validation Failed". After receiving a NA(EARO) with a status of "Validation Failed", the registering node SHOULD try and alternate Crypto-Type and if even Crypto-Type 0 fails, it may try to register a different address in the NS message.

6.2. NDPSO generation and verification

The signature generated by the 6LN to provide proof-of-ownership of the private-key is carried in the NDP Signature Option (NDPSO). It is generated by the 6LN in a fashion that depends on the Crypto-Type (see Table 1 in Section 8.2) chosen by the 6LN as follows:

* Form the message to be signed, by concatenating the following byte-strings in the order listed:

    1. The 128-bit Message Type tag [RFC3972] (in network byte order). For this specification the tag is given in Section 8.1. (The tag value has been generated by the editor of this specification on random.org).
    2. the CIPO
    3. the 16-byte Target Address (in network byte order) sent in the Neighbor Solicitation (NS) message. It is the address which the 6LN is registering with the 6LR and 6LBR.
    4. NonceLR received from the 6LR (in network byte order) in the Neighbor Advertisement (NA) message. The nonce is at least 6 bytes long as defined in [RFC3971].
    5. NonceLN sent from the 6LN (in network byte order). The nonce is at least 6 bytes long as defined in [RFC3971].
    6. 1-byte Option Length of the EARO containing the Crypto-ID.

* Apply the signature algorithm specified by the Crypto-Type using the private key.

The 6LR on receiving the NDPSO and CIPO options first checks that the EARO Length in the CIPO matches the length of the EARO. If so it regenerates the Crypto-ID based on the CIPO to make sure that the leftmost bits up to the size of the ROVR match.

If and only if the check is successful, it tries to verify the signature in the NDPSO option using the following:

* Form the message to be verified, by concatenating the following byte-strings in the order listed:

    1. The 128-bit Message Type tag given in Section 8.1 (in network byte order)
    2. the CIPO

3. the 16-byte Target Address (in network byte order) received in the Neighbor Solicitation (NS) message. It is the address which the 6LN is registering with the 6LR and 6LBR.
4. NonceLR sent in the Neighbor Advertisement (NA) message. The nonce is at least 6 bytes long as defined in [RFC3971].
5. NonceLN received from the 6LN (in network byte order) in the NS message. The nonce is at least 6 bytes long as defined in [RFC3971].
6. 1-byte EARO Length received in the CIPO.

*Verify the signature on this message with the public-key in the CIPO and the locally computed values using the signature algorithm specified by the Crypto-Type. If the verification succeeds, the 6LR propagates the information to the 6LBR using a EDAR/EDAC flow.

*Due to the first-come/first-serve nature of the registration, if the address is not registered to the 6LBR, then flow succeeds and both the 6LR and 6LBR add the state information about the Crypto-ID and Target Address being registered to their respective abstract database.

6.3. Multihop Operation

A new 6LN that joins the network auto-configures an address and performs an initial registration to a neighboring 6LR with an NS message that carries an Address Registration Option (EARO) [RFC8505].

In a multihop 6LoWPAN, the registration with Crypto-ID is propagated to 6LBR as shown in Figure 7, which illustrates the registration flow all the way to a 6LowPAN Backbone Router (6BBR) [BACKBONE-ROUTER].

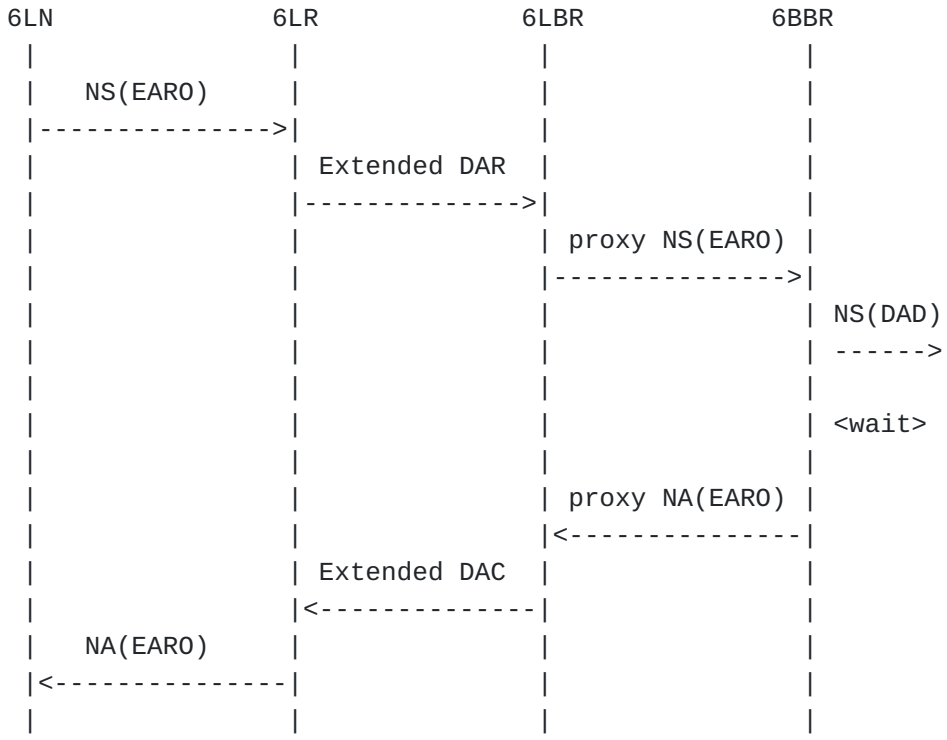

Figure 7: (Re-)Registration Flow

The 6LR and the 6LBR communicate using ICMPv6 Extended Duplicate Address Request (EDAR) and Extended Duplicate Address Confirmation (EDAC) messages [RFC8505] as shown in Figure 7. This specification extends EDAR/EDAC messages to carry cryptographically generated ROVR.

The assumption is that the 6LR and the 6LBR maintain a security association to authenticate and protect the integrity of the EDAR and EDAC messages, so there is no need to propagate the proof of ownership to the 6LBR. The 6LBR implicitly trusts that the 6LR performs the verification when the 6LBR requires it, and if there is no further exchange from the 6LR to remove the state, that the verification succeeded.

7. Security Considerations

7.1. Brown Field

Only 6LRs that are upgraded to this specification are capable to challenge a registration and repel an attack. In a brown (mixed) network, an attacker may attach to a legacy 6LR and fool the 6LBR. So even if the "A" flag could be set at any time to test the protocol operation, the security will only be effective when the all the 6LRs are upgraded.

7.2. Inheriting from RFC 3971

Observations regarding the following threats to the local network in [RFC3971] also apply to this specification.

Neighbor Solicitation/Advertisement Spoofing: Threats in section 9.2.1 of RFC3971 apply. AP-ND counters the threats on NS(EARO) messages by requiring that the NDP Signature and CIPO options be present in these solicitations.

Duplicate Address Detection DoS Attack: Inside the LLN, Duplicate Addresses are sorted out using the ROVR, which differentiates it from a movement. A different ROVR for the same Registered address entails a rejection of the second registration [RFC8505]. DAD coming from the backbone are not forwarded over the LLN, which provides some protection against DoS attacks inside the resource-constrained part of the network. Over the backbone, the EARO option is present in NS/NA messages. This protects against misinterpreting a movement for a duplication, and enables the backbone routers to determine which one has the freshest registration [RFC8505] and is thus the best candidate to validate the registration for the device attached to it [BACKBONE-ROUTER]. But this specification does not guarantee that the backbone router claiming an address over the backbone is not an attacker.

Router Solicitation and Advertisement Attacks: This specification does not change the protection of RS and RA which can still be protected by SEND.

Replay Attacks A nonce should never repeat for a single key, but nonces do not need to be unpredictable for secure operation. Using nonces (NonceLR and NonceLN) generated by both the 6LR and 6LN ensure a contributory behavior that provides an efficient protection against replay attacks of the challenge/response flow. The quality of the protection by a random nonce depends on the random number generator and its parameters (e.g., sense of time).

Neighbor Discovery DoS Attack: A rogue node that managed to access the L2 network may form many addresses and register them using AP-ND. The perimeter of the attack is all the 6LRs in range of the attacker. The 6LR MUST protect itself against overflows and reject excessive registration with a status 2 "Neighbor Cache Full". This effectively blocks another (honest) 6LN from registering to the same 6LR, but the 6LN may register to other 6LRs that are in its range but not in that of the rogue.

7.3. Related to 6LoWPAN ND

The threats and mediations discussed in 6LoWPAN ND [RFC6775] [RFC8505] also apply here, in particular denial-of-service attacks against the registry at the 6LR or 6LBR.

Secure ND [RFC3971] forces the IPv6 address to be cryptographic since it integrates the CGA as the IID in the IPv6 address. In contrast, this specification saves about 1Kbyte in every NS/NA message. Also, this specification separates the cryptographic identifier from the registered IPv6 address so that a node can have more than one IPv6 address protected by the same cryptographic identifier.

With this specification the 6LN can freely form its IPv6 address(es) in any fashion, thereby enabling either 6LoWPAN compression for IPv6 addresses that are derived from Layer-2 addresses, or temporary addresses, e.g., formed pseudo-randomly and released in relatively short cycles for privacy reasons [RFC8064][RFC8065], that cannot be compressed.

This specification provides added protection for addresses that are obtained following due procedure [RFC8505] but does not constrain the way the addresses are formed or the number of addresses that are used in parallel by a same entity. A rogue may still perform denial-of-service attack against the registry at the 6LR or 6LBR, or attempt to deplete the pool of available addresses at Layer-2 or Layer-3.

7.4. Compromised 6LR

This specification distributes the challenge and its validation at the edge of the network, between the 6LN and its 6LR. This protects against DOS attacks targeted at that central 6LBR. This also saves back and forth exchanges across a potentially large and constrained network.

The downside is that the 6LBR needs to trust the 6LR for performing the checking adequately, and the communication between the 6LR and the 6LBR must be protected to avoid tempering with the result of the test.

If a 6LR is compromised, and provided that it knows the ROVR field used by the real owner of the address, the 6LR may pretend that the owner has moved, is now attached to it and has successfully passed the Crpto-ID validation. The 6LR may then attract and inject traffic at will on behalf of that address or let a rogue take ownership of the address.

7.5. ROVR Collisions

A collision of Registration Ownership Verifiers (ROVR) (i.e., the Crypto-ID in this specification) is possible, but it is a rare event. Assuming in the calculations/discussion below that the hash used for calculating the Crypto-ID is a well-behaved cryptographic hash and thus that random collisions are the only ones possible, the formula (birthday paradox) for calculating the probability of a collision is 1 - e^{-p^2/(2n)} where n is the maximum population size (2^64 here, 1.84E19) and p is the actual population (number of nodes, assuming one Crypto-ID per node).

If the Crypto-ID is 64-bits (the least possible size allowed), the chance of a collision is 0.01% for network of 66 million nodes. Moreover, the collision is only relevant when this happens within one stub network (6LBR). In the case of such a collision, a third party node would be able to claim the registered address of an another legitimate node, provided that it wishes to use the same address. To prevent address disclosure and avoid the chances of collision on both the ROVR and the address, it is RECOMMENDED that nodes do not derive the address being registered from the ROVR.

7.6. Implementation Attacks

The signature schemes referenced in this specification comply with NIST [FIPS186-4] or Crypto Forum Research Group (CFRG) standards [RFC8032] and offer strong algorithmic security at roughly 128-bit security level. These signature schemes use elliptic curves that were either specifically designed with exception-free and constant-time arithmetic in mind [RFC7748] or where one has extensive implementation experience of resistance to timing attacks [FIPS186-4].

However, careless implementations of the signing operations could nevertheless leak information on private keys. For example, there are micro-architectural side channel attacks that implementors should be aware of [breaking-ed25519]. Implementors should be particularly aware that a secure implementation of Ed25519 requires a protected implementation of the hash function SHA-512, whereas this is not required with implementations of the hash function SHA-256 used with ECDSA256 and ECDSA25519.

7.7. Cross-Algorithm and Cross-Protocol Attacks

The keypair used in this specification can be self-generated and the public key does not need to be exchanged, e.g., through certificates, with a third party before it is used.

New keypairs can be formed for new registration as the node desires. On the other hand, it is safer to allocate a keypair that is used

only for the address protection and only for one instantiation of the signature scheme (which includes choice of elliptic curve domain parameters, used hash function, and applicable representation conventions).

The same private key MUST NOT be reused with more than one instantiation of the signature scheme in this specification. The same private key MUST NOT be used for anything other than computing NDPSO signatures per this specification.

ECDSA shall be used strictly as specified in [FIPS186-4]. In particular, each signing operation of ECDSA MUST use randomly generated ephemeral private keys and MUST NOT reuse these ephemeral private keys accross signing operations. This precludes the use of deterministic ECDSA without a random input for determination of 'k', which is deemed dangerous for the intended applications this document aims to serve.

7.8. Public Key Validation

Public keys contained in the CIPO field (which are used for signature verification) shall be verified to be correctly formed, by checking that this public key is indeed a point of the elliptic curve indicated by the Crypto-Type and that this point does have the proper order.

For points used with the signature scheme Ed25519, one MUST check that this point is not a point in the small subgroup (see Appendix B.1 of [CURVE-REPR]); for points used with the signature scheme ECDSA (i.e., both ECDSA256 and ECDSA25519), one MUST check that the point has the same order as the base point of the curve in question. This is commonly called full public key validation (again, see Appendix B.1 of [CURVE-REPR]).

7.9. Correlating Registrations

The ROVR field in the EARO introduced in [RFC8505] extends the EUI-64 field of the ARO defined in [RFC6775]. One of the drawbacks of using an EUI-64 as ROVR is that an attacker that is aware of the registrations can correlate traffic for a same 6LN across multiple addresses. Section 3 of [RFC8505] indicates that the ROVR and the address being registered are decoupled. A 6LN may use a same ROVR for multiple registrations or a different ROVR per registration, and the IID must not derive from the ROVR. In theory different 6LNs could use a same ROVR as long as they do not attempt to register the same address.

The Modifier used in the computation of the Crypto-ID enables a 6LN to build different Crypto-IDs for different addresses with a same keypair. Using that facility improves the privacy of the 6LN as the

expense of storage in the 6LR, which will need to store multiple CIPOs that contain the same public key. Note that if the attacker is the 6LR, then the Modifier alone does not provide a protection, and the 6LN would need to use different keys and MAC addresses in an attempt to obfuscate its multiple ownership.

8. IANA considerations

8.1. CGA Message Type

This document defines a new 128-bit value of a Message Type tag under the CGA Message Type [RFC3972] name space: 0x8701 55c8 0cca dd32 6ab7 e415 f148 84d0.

8.2. Crypto-Type Subregistry

IANA is requested to create a new subregistry "Crypto-Type Subregistry" in the "Internet Control Message Protocol version 6 (ICMPv6) Parameters". The registry is indexed by an integer in the interval 0..255 and contains an Elliptic Curve, a Hash Function, a Signature Algorithm, Representation Conventions, Public key size, and Signature size, as shown in Table 1, which together specify a signature scheme (and which are fully specified in Appendix B).

The following Crypto-Type values are defined in this document:

Crypto-Type value	**0 (ECDSA256)**	**1 (Ed25519)**	**2 (ECDSA25519)**
Elliptic curve	NIST P-256 [FIPS186-4]	Curve25519 [RFC7748]	Curve25519 [RFC7748]
Hash function	SHA-256 [RFC6234]	SHA-512 [RFC6234]	SHA-256 [RFC6234]
Signature algorithm	ECDSA [FIPS186-4]	Ed25519 [RFC8032]	ECDSA [FIPS186-4]
Representation conventions	Weierstrass, (un)compressed, MSB/msb first, [RFC7518]	Edwards, compressed, LSB/lsb first, [RFC8037]	Weierstrass, (un)compressed, MSB/msb first, [CURVE-REPR]
Public key size	33/65 bytes (compressed/ uncompressed)	32 bytes (compressed)	33/65 bytes (compressed/ uncompressed)
Signature size	64 bytes	64 bytes	64 bytes
Defining specification	This_RFC	This_RFC	This_RFC

Table 1: Crypto-Types

New Crypto-Type values providing similar or better security may be defined in the future.

Assignment of new values for new Crypto-Type MUST be done through IANA with either "Specification Required" or "IESG Approval" as defined in BCP 26 [RFC8126].

8.3. IPv6 ND option types

This document registers two new ND option types under the subregistry "IPv6 Neighbor Discovery Option Formats":

Option Name	Suggested Value	Reference
NDP Signature Option (NDPSO)	38	This document
Crypto-ID Parameters Option (CIPO)	39	This document

Table 2: New ND options

8.4. New 6LoWPAN Capability Bit

IANA is requested to make additions to the Subregistry for "6LoWPAN Capability Bits" created for [RFC7400] as follows:

Capability Bit	Description	Document
09	AP-ND Enabled (1 bit)	This_RFC

Table 3: New 6LoWPAN Capability Bit

9. Acknowledgments

Many thanks to Charlie Perkins for his in-depth review and constructive suggestions. The authors are also especially grateful to Robert Moskowitz and Benjamin Kaduk for their comments and discussions that led to many improvements. The authors wish to also thank Shwetha Bhandari for actively shepherding this document and Roman Danyliw, Alissa Cooper, Mirja Kuhlewind, Eric Vyncke, Vijay Gurbani, Al Morton, and Adam Montville for their constructive reviews during the IESG process. Finally Many thanks to our INT area ADs, Suresh Krishnan and then Erik Kline, who supported us along the whole process.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <https://www.rfc-editor.org/info/rfc7400>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

[FIPS186-4] FIPS 186-4, "Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4", US Department of Commerce/National Institute of Standards and Technology , July 2013.

[SEC1] SEC1, "SEC 1: Elliptic Curve Cryptography, Version 2.0", Standards for Efficient Cryptography , June 2009.

11. Informative references

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[BCP 106] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <https://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <https://www.rfc-editor.org/info/rfc7039>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[BCP 201] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms",

BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <https://www.rfc-editor.org/info/rfc7696>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption (JOSE)", RFC 8037, DOI 10.17487/RFC8037, January 2017, <https://www.rfc-editor.org/info/rfc8037>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <https://www.rfc-editor.org/info/rfc8065>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[BACKBONE-ROUTER] Thubert, P., Perkins, C., and E. Levy-Abegnoli, "IPv6 Backbone Router", Work in Progress, Internet-Draft, draft-ietf-6lo-backbone-router-20, 23 March 2020, <https://tools.ietf.org/html/draft-ietf-6lo-backbone-router-20>.

[CURVE-REPR] Struik, R., "Alternative Elliptic Curve Representations", Work in Progress, Internet-Draft, draft-ietf-lwig-curve-representations-09, 9 March 2020, <https://tools.ietf.org/html/draft-ietf-lwig-curve-representations-09>.

[breaking-ed25519] Samwel, N., Batina, L., Bertoni, G., Daemen, J., and R. Susella, "Breaking Ed25519 in WolfSSL", Cryptographers' Track at the RSA Conference , 2018, <https://link.springer.com/chapter/10.1007/978-3-319-76953-0_1>.

Appendix A. Requirements Addressed in this Document

In this section we state requirements of a secure neighbor discovery protocol for low-power and lossy networks.

*The protocol MUST be based on the Neighbor Discovery Optimization for Low-power and Lossy Networks protocol defined in [RFC6775]. RFC6775 utilizes optimizations such as host-initiated interactions for sleeping resource-constrained hosts and elimination of multicast address resolution.

*New options to be added to Neighbor Solicitation messages MUST lead to small packet sizes, especially compared with existing protocols such as SEcure Neighbor Discovery (SEND). Smaller packet sizes facilitate low-power transmission by resource-constrained nodes on lossy links.

*The support for this registration mechanism SHOULD be extensible to more LLN links than IEEE 802.15.4 only. Support for at least the LLN links for which a 6lo "IPv6 over foo" specification exists, as well as Low-Power Wi-Fi SHOULD be possible.

*As part of this extension, a mechanism to compute a unique Identifier should be provided with the capability to form a Link Local Address that SHOULD be unique at least within the LLN connected to a 6LBR.

*The Address Registration Option used in the ND registration SHOULD be extended to carry the relevant forms of Unique Interface Identifier.

*The Neighbor Discovery should specify the formation of a site-local address that follows the security recommendations from [RFC7217].

Appendix B. Representation Conventions

B.1. Signature Schemes

The signature scheme ECDSA256 corresponding to Crypto-Type 0 is ECDSA, as specified in [FIPS186-4], instantiated with the NIST prime curve P-256, as specified in Appendix B of [FIPS186-4], as specified in [RFC6234], where points of this NIST curve are represented as points of a short-Weierstrass curve (see [FIPS186-4]) and are encoded as octet strings in most-significant-bit first (msb) and most-significant-byte first (MSB) order. The signature itself consists of two integers (r and s), which are each encoded as fixed-size octet strings in most-significant-bit first and most-significant-byte first order. The hash function is SHA-256. For details on ECDSA, see [FIPS186-4]; for details on the encoding of public keys, see Appendix B.3; for details on the signature encoding, see Appendix B.2.

The signature scheme Ed25519 corresponding to Crypto-Type 1 is EdDSA, as specified in [RFC8032], instantiated with the Montgomery curve Curve25519, as specified in [RFC7748]. , where points of this Montgomery curve are represented as points of the corresponding twisted Edwards curve Edwards25519 (see Appendix B.4) and are encoded as octet strings in least-significant-bit first (lsb) and

least-significant-byte first (LSB) order. The associated hash algorithm, used internally by Ed25519 but not part of the signature process, is SHA-512, as specified in [RFC6234]. The signature itself consists of a bit string that encodes a point of this twisted Edwards curve, in compressed format, and an integer encoded in least-significant-bit first and least-significant-byte first order. For details on EdDSA, the encoding of public keys and that of signatures, see the specification of pure Ed25519 in [RFC8032].

The signature scheme ECDSA25519 corresponding to Crypto-Type 2 is ECDSA, as specified in [FIPS186-4], instantiated with the Montgomery curve Curve25519, as specified in [RFC7748], and the hash function SHA-256, as specified in [RFC6234], where points of this Montgomery curve are represented as points of the corresponding short-Weierstrass curve Wei25519 (see Appendix B.4) and are encoded as octet strings in most-significant-bit first and most-significant-byte first order. The signature itself consists of a bit string that encodes two integers, each encoded as fixed-size octet strings in most-significant-bit first and most-significant-byte first order. The hash function is SHA-256. For details on ECDSA, see [FIPS186-4]; for details on the encoding of public keys, see Appendix B.3; for details on the signature encoding, see Appendix B.2

B.2. Representation of ECDSA Signatures

With ECDSA, each signature is an ordered pair (r, s) of integers [FIPS186-4], where each integer is represented as a 32-octet string according to the Field Element to Octet String conversion rules in [SEC1] and where the ordered pair of integers is represented as the rightconcatenation of these representation values (thereby resulting in a 64-octet string). The inverse operation checks that the signature is a 64-octet string and represents the left-side and right-side halves of this string (each a 32-octet string) as the integers r and s, respectively, using the Octet String to Field Element conversion rules in [SEC1].

B.3. Representation of Public Keys Used with ECDSA

ECDSA is specified to be used with elliptic curves in short-Weierstrass form. Each point of such a curve is represented as an octet string using the Elliptic Curve Point to Octet String conversion rules in [SEC1], where point compression may be enabled (which is indicated by the leftmost octet of this representation). The inverse operation converts an octet string to a point of this curve using the Octet String to Elliptic Curve Point conversion rules in [SEC1], whereby the point is rejected if this is the so-called point at infinity. (This is the case if the input to this inverse operation is an octet string of length 1.)

B.4. Alternative Representations of Curve25519

The elliptic curve Curve25519, as specified in [RFC7748], is a so-called Montgomery curve. Each point of this curve can also be represented as a point of a twisted Edwards curve or as a point of an elliptic curve in short-Weierstrass form, via a coordinate transformation (a so-called isomorphic mapping). The parameters of the Montgomery curve and the corresponding isomorphic curves in twisted Edwards curve and short-Weierstrass form are as indicated below. Here, the domain parameters of the Montgomery curve Curve25519 and of the twisted Edwards curve Edwards25519 are as specified in [RFC7748]; the domain parameters of the elliptic curve Wei25519 in short-Weierstrass curve comply with Section 6.1.1 of [FIPS186-4]. For further details on these curves and on the coordinate transformations referenced above, see [CURVE-REPR].

General parameters (for all curve models):

p 2^{255}-19
(=0x7fffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffed)
h 8
n
72370055773322622139731865630429942408571163593799076060019509382854542509 89
(=2^{252} + 0x14def9de a2f79cd6 5812631a 5cf5d3ed)

Montgomery curve-specific parameters (for Curve25519):

A 486662
B 1
Gu 9 (=0x9)
Gv
14781619447589544791020593568409986887264606134616475288964881837755586237401
(=0x20ae19a1 b8a086b4 e01edd2c 7748d14c 923d4d7e 6d7c61b2
29e9c5a2 7eced3d9)

Twisted Edwards curve-specific parameters (for Edwards25519):

a -1 (-0x01)
d -121665/121666
(=37095705934669439343138083508754565189542113879843219016388785533085940283555)
(=0x52036cee 2b6ffe73 8cc74079 7779e898 00700a4d 4141d8ab
75eb4dca 135978a3)

Gx

151122213495354007725011514095885315114540126930418572060461132839498477622021

(=0x216936d3 cd6e53fe c0a4e231 fdd6dc5c 692cc760 9525a7b2 c9562d60 8f25d51a)

Gy 4/5

(=463168356949264781694283940034751631413079938662562256157830336031652518559600)

(=0x66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666658)

Weierstrass curve-specific parameters (for Wei25519):

a

19298681539552699237261830834781317975544997444273427339909597334573241639236

(=0x2aaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaa98 4914a144)

b

55751746669818908907645289078257140818241103727901012315294400837956729358436

(=0x7b425ed0 97b425ed 097b425e d097b425 ed097b42 5ed097b4 260b5e9c 7710c864)

GX

19298681539552699237261830834781317975544997444273427339909597334652188435546

(=0x2aaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaaaaaaa aaad245a)

GY

14781619447589544791020593568409986887264606134616475288964881837755586237401

(=0x20ae19a1 b8a086b4 e01edd2c 7748d14c 923d4d7e 6d7c61b2 29e9c5a2 7eced3d9)

Authors' Addresses

Pascal Thubert (editor)
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
06254 MOUGINS - Sophia Antipolis
France

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Behcet Sarikaya

Email: sarikaya@ieee.org

Mohit Sethi
Ericsson
FI-02420 Jorvas
Finland

Email: mohit@piuha.net

Rene Struik
Struik Security Consultancy

Email: rstruik.ext@gmail.com